AugAuth : Shoulder-Surfing Resistant Authentication for Augmented Reality

Ruide Zhang[†], Ning Zhang[†], Changlai Du[†], Wenjing Lou[†], Y. Thomas Hou[†], and Yuichi Kawamoto[‡]

[†]Virginia Polytechnic Institute and State University, VA, USA
[†]{rdzhang,ningzh,leondu,wjlou,thou}@vt.edu
[‡]Tohoku University, Sendai, Japan
[‡]youpsan@it.ecei.tohoku.ac.jp

***Abstract*—As computing system continues to play an increasing role in daily life, user authentication is now an important component. One of the most widely accepted methods for user authentication is through proof of knowledge of a piece of secret information, such as password. However, entering this non-mutable secret for authentication in public space often allows attackers to steal the secret by shoulder surfing or video recording.**

We observe that it is possible to block attacker's access to user input using augmented reality (AR) display, which is only available to the user. Based on this intuition, we present AugAuth, an authentication scheme in AR using commercial off-the-shelf(COTS) gesture control sensors as an input device. AugAuth can resist against shoulder surfing by presenting user input interface that is only visible to the user and is unique every time. To enable user input with finger movement using the gesture control armband, Myo, we have solved several challenges in electromyogram signal processing, such as annotating the start of signal and finger classification. The experiment results for our input system of a group of volunteers show that our finger classification function has high accuracy and AugAuth is practical for use in real life authentication scenarios.

I. Introduction

With the recent advancement in embedded devices and widespread network connectivity, mobile devices are playing an increasing role in our daily life. Despite decades of research, it remains an active research topic to provide secure and usable user authentication for these handheld devices. One of the most widely accepted methods for authentication is by demonstrating ownership of a piece of secret information, such as password or pin.

However, using knowledge of a piece of secret information for authentication has its own drawbacks, especially when such secret is entered in public space. The confidentiality of user secret is often leaked to malicious attackers via shoulder surfing [1], security camera recordings [2] or malicious image processing unit on wearable devices [3], [4]. In some cases, the attacker doesn't even need to see the screen of the user's tablet, the pin entered can be inferred with just the position of fingers relative to the tablet [5] or from the reflecting of screen on the retina of users [6]. To mitigate these risks, there has been a large number of research focusing on development of different types of password-equivalent secrets [7], [8], [9] while others rely on biometrics [10]. With the recent development of human-computer interfacing and battery technology, we are now at the uprising frontier of augmented reality.

In this paper, we present AugAuth, an innovative authentication scheme with unique features enabled by AR headset and gesture control device. The design of AugAuth is based on the observation that user secrets are leaked to attackers because an attacker can observe the activities of the user (i.e. the input sequence) and also the input device screen. However, if we display the authentication interface only to the user and nobody else through AR, an attacker will lose the access to the input device screen. Furthermore, the orientation or layout of the authentication interface (i.e. the virtual keyboard) can be randomized each time it is used to prevent replay attack.

In AugAuth, users are presented with a virtual keypad with numbers in randomized order inside the AR view that is only available to the user. AugAuth employs a gesture control device to capture the finger movement as user input on such virtual keypad. This way, even if the eavesdroppers are capable of observing the activities of the user, they would have no way of knowing what the password is. Despite the simplicity of the approach, capturing the finger movement with gesture control device in a simple-to-use manner remains a challenging issue using commodity device. It is unrealistic to put sensors on every finger of the user due to usability concern. We chose to use one of the most popular gesture control armband, Myo [11], to capture finger movement as user input. Myo is an armband equipped with eight electromyogram sensors. It is designed to detect simple hand gestures such as holding the fist or completely opening up the hand, our goal on the other hand is to take the coarse-grained information provided by the armband to predict finger movement.

To implement accurate finger detection with Myo we have to tackle three major challenges. First, it is difficult to determine the start of the input events because there exists lots of noise caused by user's irregular moving. Second, the pushing and releasing of finger when a user is performing an input activity will produce two overlapping electromyography (EMG) signals which is hard to process with. Third, the relationship between EMG signal and finger movement is not clear.

Our approach analyzes the time and frequency characteristics of the signals for finger movement events to build up AugAuth system with two subsystems, input event detection subsystem and finger movement classification subsystem. In our scheme, by putting EMG signal through input event detection subsystem, we obtain the exact time stamp of when a user has finger movement. And by placing EMG signal around the time stamp into the finger movement classification subsystem, we are able to infer the exact finger a user is moving.

We summarize our main contributions as follows:

- We propose a novel password authentication scheme that is resistant to shoulder surfing or camera recording by exploiting the unique private display feature in AR.
- We develop new algorithms to capture the dynamics of EMG signals in finger movement .
- We demonstrate the feasibility of our approach by showing it is possible to capture finger movements with just a commercial off-the-shelf gesture control equipment, Myo. We also present our study on the finger movement detection accuracy on a group of volunteers.

II. Background

In our proposed scheme, it includes two devices, an augmented reality headset such as hololens [12] and a gesture control device. The headset will serve as a display device and to detect user input depends on the gesture control device. The challenging part in our scheme sits on how to detect fine-grained finger movement with gesture control device. In this paper, we launch proof-of-concept experiment with COTS device, Myo, which costs 169 dollars. Myo [11] is a gesture control device that is designed to be worn on the arm of a user. It's light-weighted with only 93 grams. The gesture control device is equipped with multiple sensors to provide seamless human-computer interaction. Myo is connected to computer desktop or mobile devices using bluetooth. It is powered by an ARM Cotex M4 processor which is very energy efficient. With one charge, the arm band can be used for a full day. Within the slick design, it houses highly sensitive medical grade sensors including eight EMG sensors. EMG signal is generated according to body movement and can be detect by tiny devices called electrodes on human surface [13]. Some samples of EMG signals are shown in Fig. 1, which are recorded when a user is performing finger taps.

III. Threat Model and Assumptions

Authentication by a password occurs in different settings and with different applications including online banking, gaming, online medical record, email in areas with increased security requirements. We model password authentication more abstractly as a game between three parties: a machine interrogator, a human oracle, and a human observer (or video recorder) [14].

The objective of the oracle is to authenticate himself to the interrogator by his password. The objective of the interrogator is to decide whether the oracle knows the correct password by asking the oracle questions. The observer observes all activities performed by the oracle but he is not able to observe the interrogator interface; his objective is to impersonate the oracle in subsequent games with the same interrogator. The game also assumes the oracle and interrogator shares a password by which the interrogator can verify whether the oracle's input is correct and matches his identity.

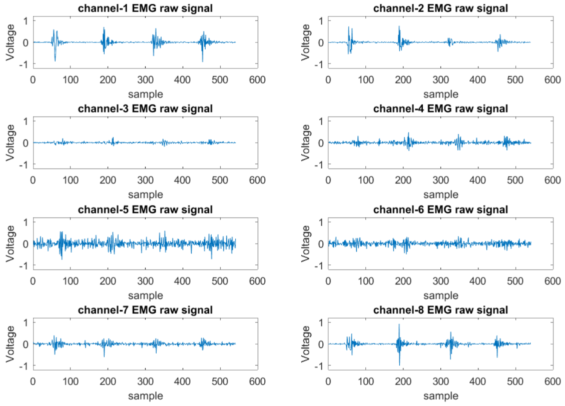

Fig. 1: Diagram of real-life EMG signals collected by Myo

We assume that the observer cannot verify the correctness of a given password unless he also knows the shared secret (and we assume he does not). Additionally, we expect that the interrogator keeps a record of how often an oracle successively inputs a false password. If the count reaches three, the interrogator voids the oracle's authorization until the oracle waits another thirty minutes. We consider the case that the observer has the capability to record all the activities of oracle when he is interacting with the interrogator without error for every single game. Having explained our assumptions and threat model, we continue by describing our AugAuth methods.

IV. AugAuth: Augmented Reality Authentication

A. AugAuth Design Model

When a user is putting on Myo for the first time, he will be instructed by the headset to perform multiple finger taps with different fingers on any surface which will generate labeled sensor data for the system. With the labeled sensor data collected, a model will be built using supervised machine learning technique and every model corresponds to specific user. After the model is generated, the user will be instructed to set up password. Then, the augmented reality headset will provide the user with an virtual keyboard in the user's view. The virtual keyboard is composed of eight distinct randomized numbers from 0 to 7, for example, the key can be arranged as 57612304. And the user can use finger tap to input the initial password which completes the initialization phase.

During the authentication phase, the AR headset again presents the user a virtual keyboard but with different keyboard layout in the user's AR view. The user needs to use finger taps to select the correct password during the initialization phase. Otherwise, the system will reject the user.

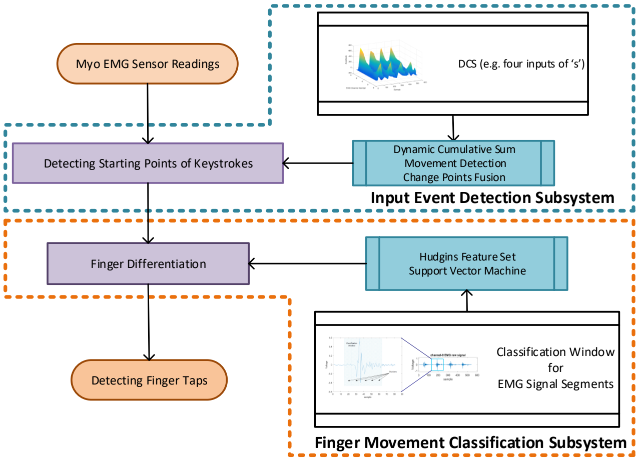

Fig. 2: System framework of AugAuth.

B. EMG Signal Modeling

In AugAuth, accurate detection and classification of finger taps is of most importance and it is based on the analysis and characterization of forearm EMG signal during the user input event. When recorded at a cellular level by EMG sensor inside Myo, forearm EMG signal can be viewed as a series of action potentials, and its time and frequency characteristics is related to the phase of the mechanical activity. In our case, the recorded electromyographic signals can be modeled by a random process

$$x(t) = \sum_{i=1}^{n} C_i(t) + \sum_{i=1}^{n} R_i(t) + n(t) \tag{1}$$

This equation is a composite of multiple types of signals collected by the EMG sensor (activity burst, noise ...). $\sum_{i=1}^{n} C_i(t)$ are our target signals which are caused by the pushing actions while $\sum_{i=1}^{n} R_i(t)$ are caused by the releasing actions. The superscript n means the number of keystrokes performed. Both the pushing and releasing actions follow a pattern of short potentials which appear with the acts of fingers. At last, $n(t)$ is the white noise caused by multiple factors like environmental conditions or thermal noise, and it is well known as a stationary Gaussian process.

Two specificities make the signal difficult to analyze. First, the impedance on human surface makes the EMG signal have tail. Second, because the pushing and releasing of a finger tap action are extremely close to each other which is about 100 milliseconds according to [15], the signal collected by EMG sensor is hard to distinguish. Nevertheless, our objective is to construct an algorithm which is capable of continuously detecting the exact timestamps of the pushing motions and to accurately classify the actions. The system overview of how we implement finger taps classification is as Fig. 2.

C. Input Event Detection Subsystem

In order to extract the exact timestamps of finger taps, we first construct an input event detection subsystem. In the subsystem, we first calculate the dynamic cumulative sum (DCS) of the collected EMG signals which includes technology from digital signal processing (DSP) and statistics. After that, through analyzing the DCS of the EMG signal, we can obtain the timestamp of each target action. The key insight is that, the DCS will reach maximum during the motion proved in [16]. So the each turning point of DCS represents one action. Besides, the signal noise ratio (SNR) of different EMG channels for movements of different fingers varies. Thus, we also invent a change point fusion algorithm here to improve the detection performance. What the change point fusion algorithm does is to merge events which are close to each other in different EMG channels.

1) DCS: To take advantage of DCS, we first need to make sure our EMG signals follow Gaussian distribution. We will show that in the following part of this section. DCS is based on the the local dynamic cumulative sum around the point of change t_m. Basically, DCS calculate the local cumulative sum of the likelihood ratios between the segments before and after time point t_m. Let us assume the two segments are $S_b^{(t_m)}$ (before t_m) and $S_a^{(t_m)}$ (after t_m) and the width of these two segments is W. $S_b^{t_m} : x_{i;i=t_m-W,\dots,t_m-1}$ follows a pdf $f_{\theta_b}(x_i)$ and $S_a^{t_m} : x_{i;i=t_m+1,\dots,t_m+W}$ follows a pdf $f_{\theta_a}(x_i)$. The parameters $\hat{\theta}_b$ and $\hat{\theta}_a$ are estimated using $S_b^{(t_m)}$ and $S_a^{(t_m)}$. The DCS is defined as the sum of the logarithm of likelihood ratios from the beginning of the signal to the time t_m:

$$DCS^{(t_m)}(S_a^{(t_m)}, S_b^{(t_m)}) = \sum_{i=1}^{t_m} Ln \frac{f_{\hat{\theta}_a}{}^{(t_m)}(x_i)}{f_{\hat{\theta}_b}{}^{(t_m)}(x_i)} \tag{2}$$

where, the θ can be estimated by the variance of each segments.

We further adopt wavelet transform (WT) [17] to launch multiscale decomposition of the two local segments of the EMG signal to improve the movement detection accuracy. WT is applied to both of the before and after segments and these multiscale representations combine variance information and frequency components. The choice of motherlet is a crucial point when adopting WT. In [18], they conclude the best wavelet to keep for human movement EMG signal case is the second-order Coiflet associated with the first five decomposition scales obtained by Shannon entropy criterion. The results of our experiment reinforce their conclusion.

During the development of input event detection subsystem, the assumption we have made is that the WT decompositions of EMG signals are multidimensional Gaussian. Fig. 3 presents an example of the histograms of randomly selected 600-sample and its WT decomposition at five scales. The experiment shows that our case subjects to the assumption. We can observe from Fig. 3 that the signals can be assumed to adequately follow Gaussian distributions. Similar distributions are obtained from other segments also.

Fig. 4 is the figure for all eight channels for the input of four finger taps with ring finger, and we can find that some channels perform better than others. We can also see from Fig. 4, the channel 1 and channel 8 which are next to each other

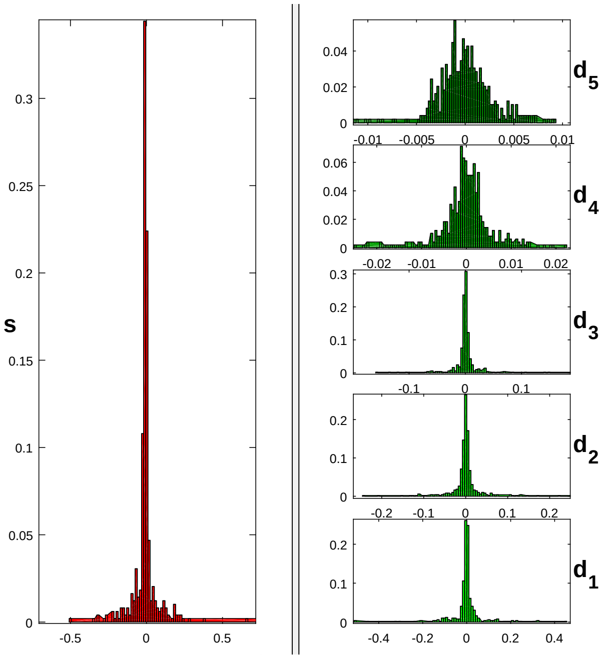

Fig. 3: EMG signals and 5-level detailed WT decompositions histograms.

perform best in this case. That is because the muscle used to do the finger tap with ring finger majorally sits near that part of the forearm.

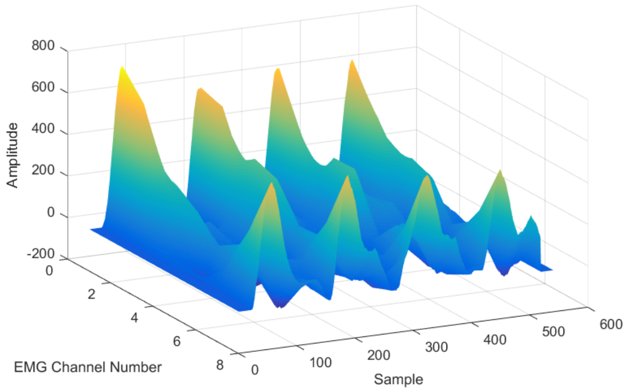

Fig. 4: DCS of all eight EMG channels.

2) Algorithms: The two algorithms to find the correct movement timestamp are described in this section. The first algorithm is the movement detection algorithm which is developed to calculate the DCS and to detect the pushing movements in the EMG signals. In the movement detection algorithm, we set up threshold T to get rid of the releasing movements. The value of T is set to 350 empirically. Then we redirect the output timestamps to the second algorithm, change point fusion algorithm. The reason why we need to fuse points of change is that no channel alone can detect all finger movements. As in Fig. 4, the movement of different fingers correlates to different portion of forearm muscle. Empirically, we adopt two distinct channels and the threshold x in the change point fusion algorithm is set to 20 which is 100 milliseconds in time scale.

Algorithm 1: movement detection algorithm

```
At each sample, the $DCS$ is calculated according to (3) using the two
segments $S_b^{t_m}: x_{i;i=t_m-W,\dots,t_m-1}$ and $S_a^{t_m}: x_{i;i=t_m+1,\dots,t_m+W}$
if The DCS has a turning point at that sample which indicates that it
may be a finger pushing movement or a finger releasing movement then
    if There is a releasing movement before then
        This is a pushing movement, record the timestamp
        Move to the next sample
    else
        if The difference between this movement and the former pushing
        movement exceeds a threshold $T$ then
            This is a releasing movement
            Move to the next sample
        else
            This is a pushing movement, record the timestamp
            Move to the next sample
        end if
    end if
else
    Move to the next sample
end if
Output the timestamps recorded
```

Algorithm 2: change point fusion algorithm

```
Get the recorded timestamps from the output of movement detection
algorithm for selected channels and sort them into list $L1$ ascendingly.
Generate an empty list $L2$
Start from the first element $l_m$ in $L1$ and do the following.
if Any timestamp from other selected channels are close to $l_m$ within
threshold $x$ then
    Add $l_m$ into list $L2$
    Delete timestamps close to $l_m$ within threshold $x$ in list $L1$
    Delete $l_m$ in list $L1$
    Go to the next element in list $L1$
else
    Go to the next element in list $L1$
end if
Output the list $L2$
```

D. *Finger Movement Classification Subsystem*

After the input event detection subsystem, we now have the timestamp for each finger action. Here we set up a window for the EMG signal at each timestamp. According to experiments, the performance of classification performs best when the size of the sliding windows is 45 samples which is 225 milliseconds. Besides, we add offset to the timestamp so that the sliding window can include the signal for the whole movement. The whole process for this section will include two parts, feature extraction and classification.

1) Feature extraction: We extract Hudgins feature set [19] from each motion. The Hudgin's time-domain features are comprised of five different features for a given classification window. Here we divide the classification window into five equally-divided segments as in Fig. 5 and each of the segments will have five features. So there will be a total of 30 features per channel (including the undivided classification window). These features include mean absolute value (MAV), difference MAV, zero crossing, slope sign changes and waveform length.

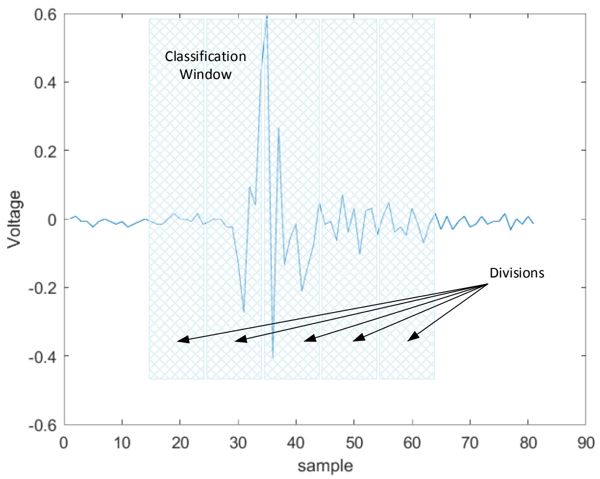

Fig. 5: Classification window and its divisions

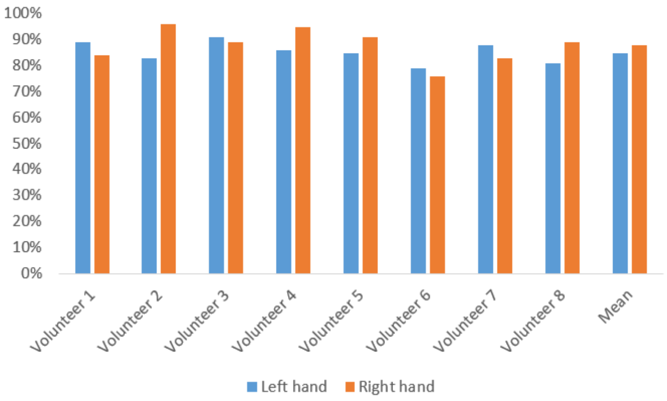

Fig. 6: Finger detection accuracy

2) Classification: We have implemented the classifier with supervised learning technique. We ask people to perform finger taps as needed to gather labeled samples. We further take advantage of the labeled samples to do a supervised learning using support vector machine (SVM) [20] classifier. The trained classifier will give us which finger the sample is related to when it is given the EMG signal for a finger tap action.

V. Experiments and results

Ideally, the experiment should be conducted to verify not only the input accuracy of Myo device, but also the usability of the interfaces after randomization. However, augmented reality headsets are not yet available. For our evaluation in this work, we focus primarily on how well simple wearable sensors are in capturing finger movements as input for authentication.

We recruit eight volunteers to conduct the experiment. In the experiment, we ask the volunteers to perform finger taps for each fingers except thumb for twenty times for training the SVM classifier, and another twenty times for testing. Meanwhile, we employ Myo to record the EMG signals on both forearms of the volunteers. All of the eight participants are between twenty and forty years old, including three women and five men.

To evaluate the performance of finger movement classifier, we define finger detection accuracy as the possibility of correct classification. The ground truth of the finger taps is recorded by us during the experiments.

We evaluate the performance of our proposed system according to finger detection accuracy. What worth mentioning here is that we adopt the movement detection algorithm to extract the timestamps and the algorithm fully detected all the movements. On the order hand, there is one SVM classifier for each volunteer trained by their own labeled samples. The finger detection accuracy for each volunteer is shown in Fig. 6.

As shown in Fig. 6, we can observe that the average accuracy is eighty-six percent. Through the experiment, we can also observe that the classifiers for volunteer from three to five have a better performance. After reviewing the video record of the experiments, we find out that the people who are using clips, an accessory of Myo, to tighten Myo tend to have a higher accuracy which is almost nighty percent. This is becuase electromyogram signals can be better captured with good skin contact. On the other hand, despite loose contact with the skin, the finger identification accuracy is still acceptable with the lowest at 80%.

According to the result of the experiment, we can not assume perfectly accurate finger classification. Therefore, it is necessary to build error tolerance into the AugAuth system. To compensate the inaccurate input error, we can view the imperfect classification as a noisy channel and adopt technology from information theory. For example, error correction coding can be appended to the user-defined pin numbers. The machine will generate parity bits according to the pin which user has set up initially. And the machine will ask user to remember both the pin and the parity bits. During authentication, the user will be instructed to input both the pin and the parity bit. As long as the number of misclassified inputs does not exceed a specific percentage, the user will be authenticated. However, this scheme may face usability issue because the user is required to remember a small number of random parity bits. Another alternative is to let the system authenticate the user as long as a percentage of entered pin numbers matches. For example, let us assume the pin has ten digits, and the pin detected by the system can have three mistakes at most. Then through simple mathematic calculation, the possibility of false positive is 98.72%. Although this will reduce the search space for brute force attack, from the usability aspect, it doesn't have the drawback of the error correction code. Nevertheless, the problem of inaccurate input originates from the commodity wearable device, due to imperfect of sensors on Myo. The concept of AugAuth does not limit the use of input device to only Myo. With the development of technology, it is possible to provide AugAuth with perfect input accuracy.

VI. Related Work

Researchers have explored and improved various methods to authenticate users for many years. Typical authentication methods consists of three main categories: (1) Token based, e.g., a security token generator (2) Biometric based, e.g., fingerprints, and (3) Knowledge based, e.g., passwords. Although biometric methods provide a high level of security, they requires costly hardware. On the other hand, security token is troublesome to use and traditional knowledge based methods suffers shoulder surfing attacks. In lieu of an alphanumeric password, researchers have examined the feasibility of other authentication schemes [21]. Recently, research has been launched on authentication for google glass [22], [23], [24]. And this paper further explores the path and provides a shoulder-surfing-proof and easy-to-use scheme.

Processing of EMG signals has also received significant attentions due to its medical applications [25], [26], [27]. However, the current simple threshold methods and simple energy comparison [25], [26], [27] are not suitable for cases where signals appear dynamic and noisy. We address this problem by building up input event detection subsystem based on DCS.

VII. Conclusion

In this work, we present AugAuth, a shoulder-surfing resistant gesture authentication for augmented reality, which is based on the observation that display in augmented reality is only shown to the user. By randomizing the authentication interface in the display, coupling with fine grained finger movement detection using COTS armband, AugAuth can provide user the ability to authenticate themselves securely even under the observation of attackers. We addresses several unique challenges in using the EMG signal from Myo to capture finger movement of user. More specifically, we invent a new movement detection algorithm based on DCS to reliably detect movement events of fingers. Furthermore, we also propose and implement the finger classification process. Through experiment with a group of volunteer, we show the feasibility of AugAuth.

Acknowledgment

This work was supported in part by the National Science Foundation under Grants CNS-1405747, CNS-1443889, CNS-1446478, CNS-1217889.

References

[1] “Shoulder surfing (computer security).” https://en.wikipedia.org/wiki/Shoulder_surfing_(computer_security), Nov 2014. [Online; posted Nov-2014].

[2] “Hackers using startling new ways to steal your passwords.” http://www.techworm.net/2015/04/hackers-using-startling-new-ways-to-steal-your-passwords.html, April 2015. [Online; posted April-2016].

[3] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu, “Friend or foe?: Your wearable devices reveal your personal pin,” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 189–200, ACM, 2016.

[4] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, “When good becomes evil: Keystroke inference with smartwatch,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1273–1285, ACM, 2015.

[5] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, “Blind recognition of touched keys on mobile devices,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1403–1414, ACM, 2014.

[6] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm, “Seeing double: Reconstructing obscured typed input from repeated compromising reflections,” in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 1063–1074, ACM, 2013.

[7] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, “Reducing shoulder-surfing by using gaze-based password entry,” in *Proceedings of the 3rd symposium on Usable privacy and security*, pp. 13–19, ACM, 2007.

[8] F. Schaub, M. Walch, B. Könings, and M. Weber, “Exploring the design space of graphical passwords on smartphones,” in *Proceedings of the Ninth Symposium on Usable Privacy and Security*, p. 11, ACM, 2013.

[9] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, “Design and evaluation of a shoulder-surfing resistant graphical password scheme,” in *Proceedings of the working conference on Advanced visual interfaces*, pp. 177–184, ACM, 2006.

[10] “The most common authentication methods used today.” http://www.tweakandtrick.com/2012/06/most-common-authentication-methods-used.html, June 2012. [Online; posted June-2016].

[11] “Myo.” https://www.myo.com/.

[12] “hololens.” https://www.microsoft.com/microsoft-hololens/en-us.

[13] M. Reaz, M. Hussain, and F. Mohd-Yasin, “Techniques of emg signal analysis: detection, processing, classification and applications,” *Biological procedures online*, vol. 8, no. 1, pp. 11–35, 2006.

[14] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[15] D. Asonov and R. Agrawal, “Keyboard acoustic emanations,” in *IEEE Symposium on Security and Privacy*, vol. 2004, pp. 3–11, 2004.

[16] M. Khalil and J. Duchêne, “Dynamic cumulative sum approach for change detection,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 47, no. 4, p. 1205, 1999.

[17] C. K. Chui, *An introduction to wavelets*, vol. 1. Academic press, 2014.

[18] Y. Al-Assaf, “Surface myoelectric signal analysis: Dynamic approaches for change detection and classification,” *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 2248–2256, Nov 2006.

[19] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, “Classification of the myoelectric signal using time-frequency based representations,” *Medical engineering & physics*, vol. 21, no. 6, pp. 431–438, 1999.

[20] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[21] Q. Sun, Z. Li, X. Jiang, and A. Kot, “An interactive and secure user authentication scheme for mobile devices,” in *2008 IEEE International Symposium on Circuits and Systems*, pp. 2973–2976, IEEE, 2008.

[22] D. K. Yadav, B. Ionascu, S. V. K. Ongole, A. Roy, and N. Memon, “Design and analysis of shoulder surfing resistant pin based authentication mechanisms on google glass,” in *International Conference on Financial Cryptography and Data Security*, pp. 281–297, Springer, 2015.

[23] D. V. Bailey, M. Dürmuth, and C. Paar, “"typing" passwords with voice recognition: How to authenticate to google glass,” in *Proc. of the Symposium on Usable Privacy and Security*, 2014.

[24] S. Ishimaru, K. Kunze, K. Kise, J. Weppner, A. Dengel, P. Lukowicz, and A. Bulling, “In the blink of an eye: Combining head motion and eye blink frequency for activity recognition with google glass,” in *Proceedings of the 5th Augmented Human International Conference*, AH ’14, (New York, NY, USA), pp. 15:1–15:4, ACM, 2014.

[25] F. H. Chan, Y.-S. Yang, F. Lam, Y.-T. Zhang, and P. A. Parker, “Fuzzy emg classification for prosthesis control,” *IEEE transactions on rehabilitation engineering*, vol. 8, no. 3, pp. 305–311, 2000.

[26] K. A. Farry, I. D. Walker, and R. G. Baraniuk, “Myoelectric teleoperation of a complex robotic hand,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 775–788, 1996.

[27] G. Tsenov, A. Zeghbib, F. Palis, N. Shoylev, and V. Mladenov, “Neural networks for online classification of hand and finger movements using surface emg signals,” in *2006 8th Seminar on Neural Network Applications in Electrical Engineering*, pp. 167–171, IEEE, 2006.